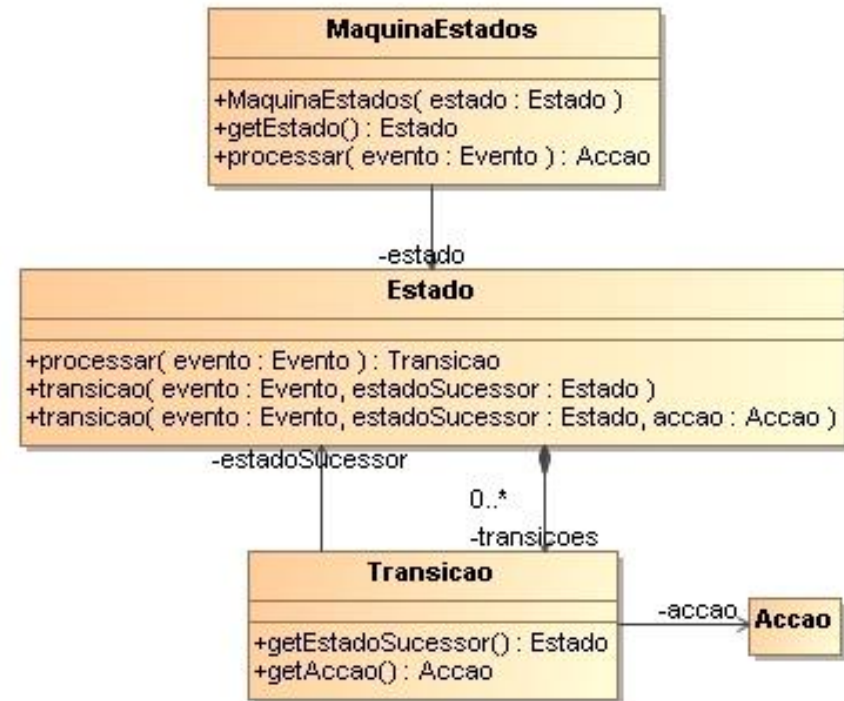


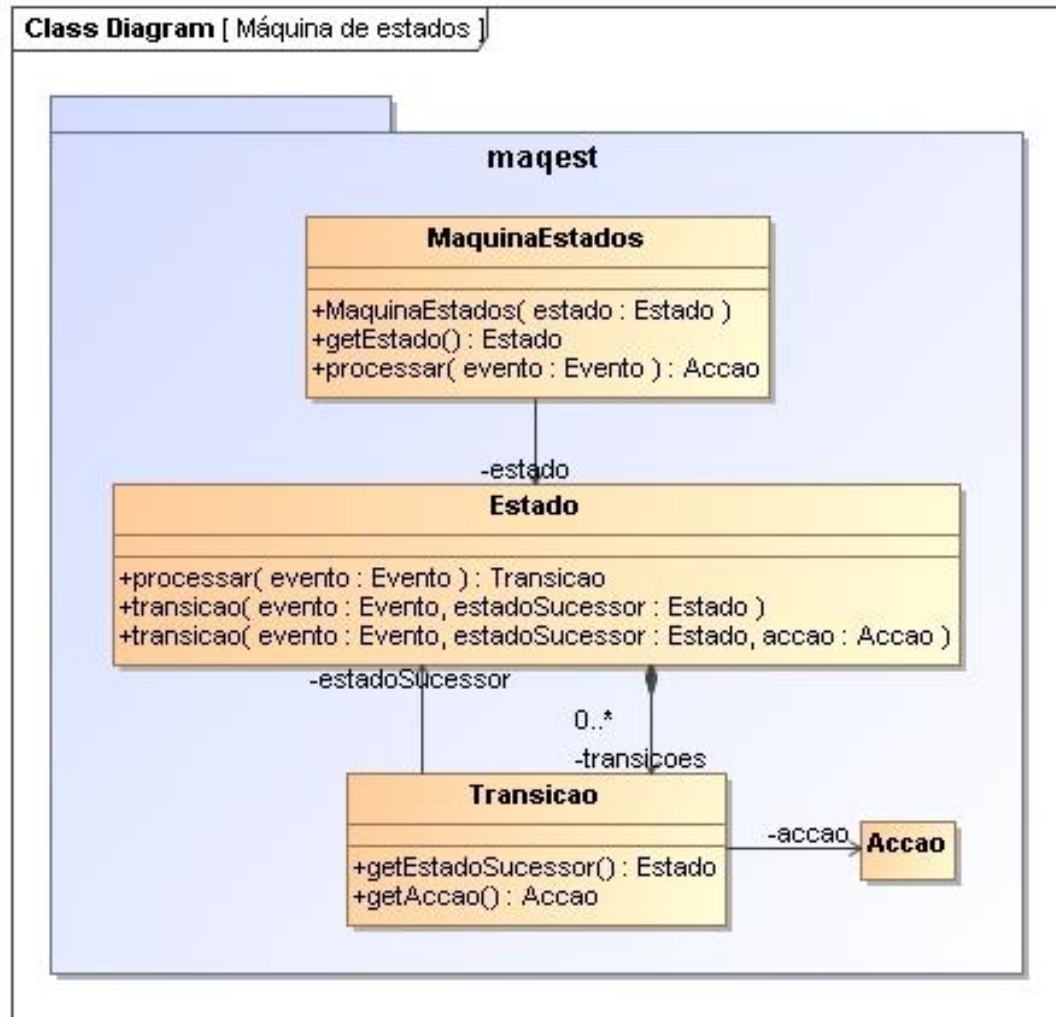
MÁQUINA DE ESTADOS

- Conceitos principais
 - Máquina de estados
 - Estado (actual)
 - Processar evento, gerando uma acção
 - Estado
 - Transições associadas a cada evento
 - Definir transição
 - Processar evento, gerando uma transição
 - Transição
 - Estado sucessor
 - Acção (de transição)



MÁQUINA DE ESTADOS

Modelo de estrutura



Problema

Como tornar a máquina de estados uma biblioteca independente da personagem para utilização em contextos gerais?

MÁQUINA DE ESTADOS GENÉRICA

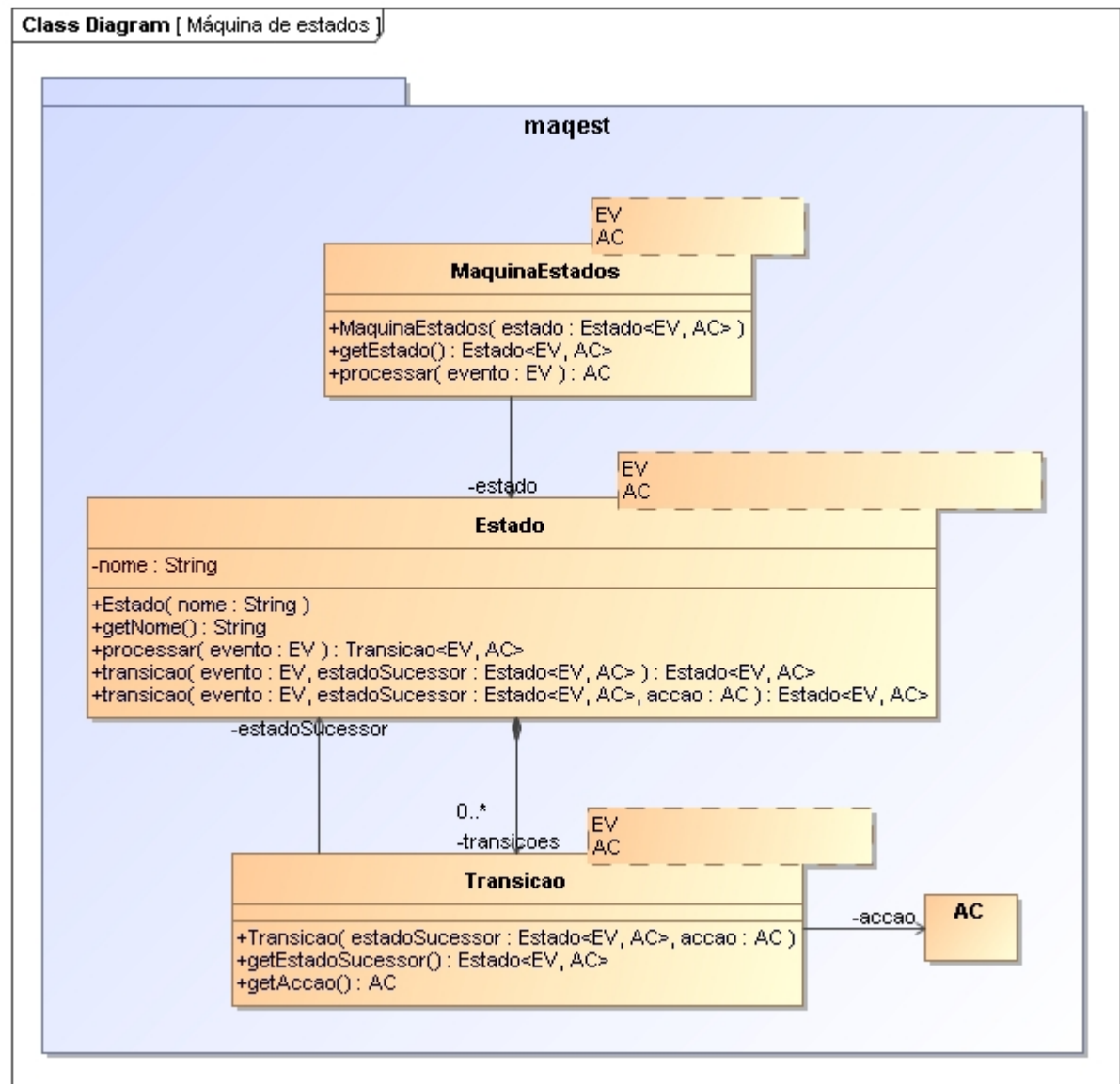
Solução

Utilização de tipos genéricos

EV – Tipo evento

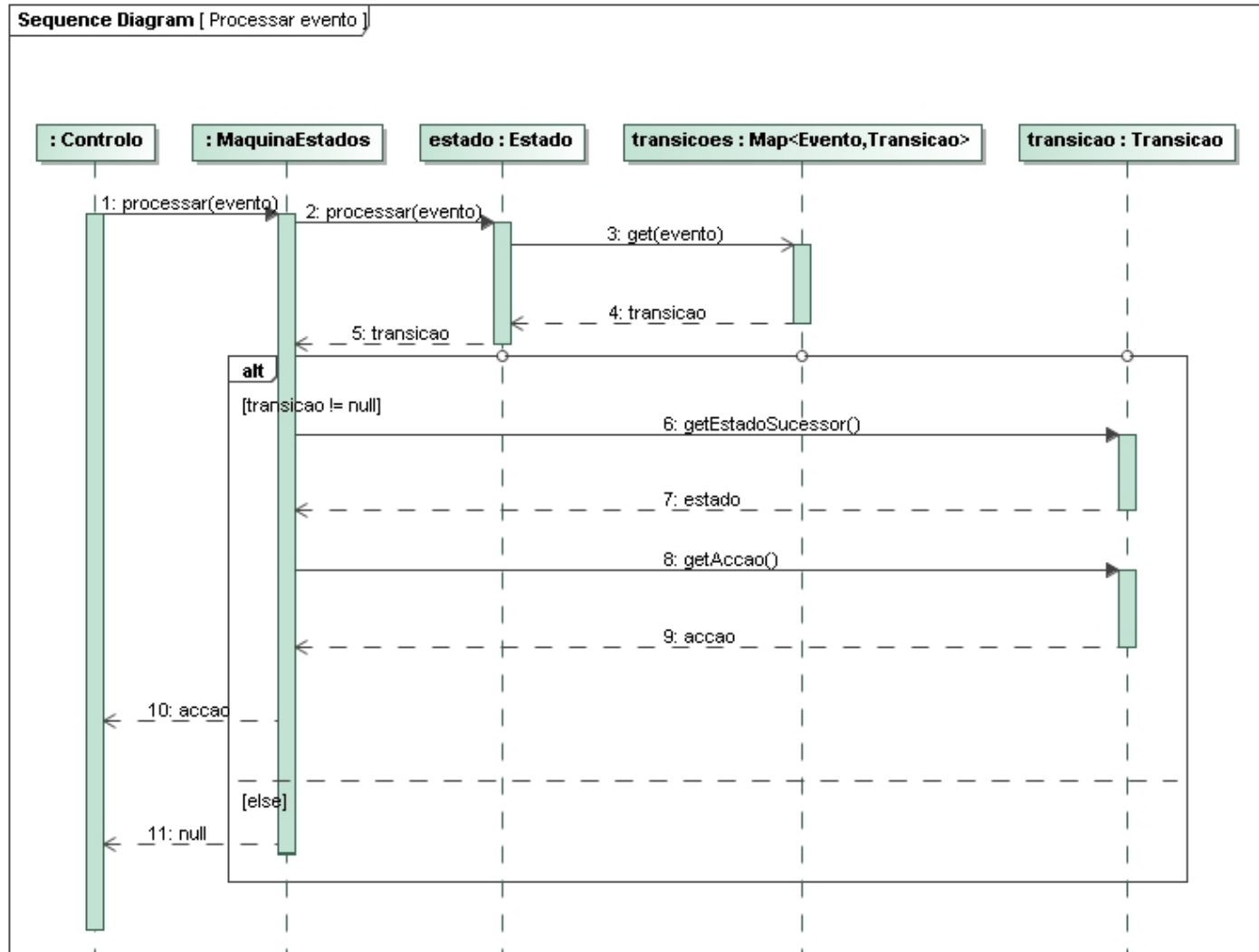
AC – Tipo acção

IMPLEMENTAÇÃO...
(Estrutural)



MÁQUINA DE ESTADOS

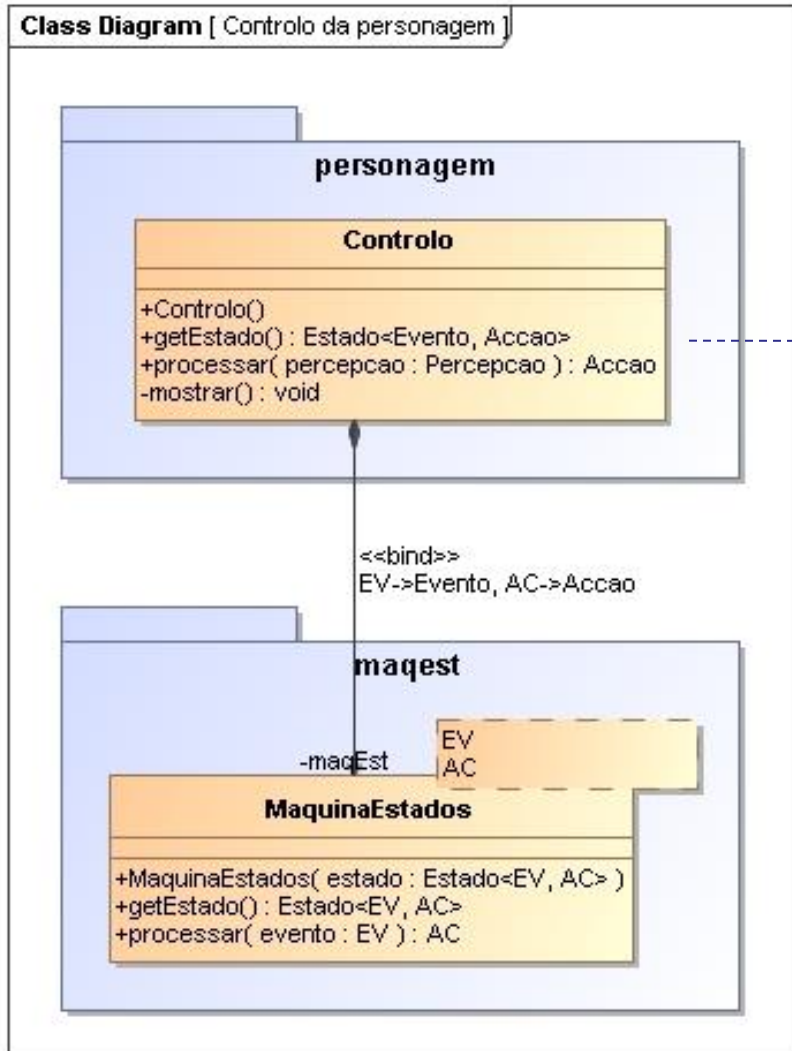
Processar evento



IMPLEMENTAÇÃO...

(Comportamental)

CONTROLO DA PERSONAGEM



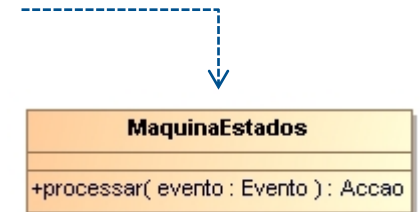
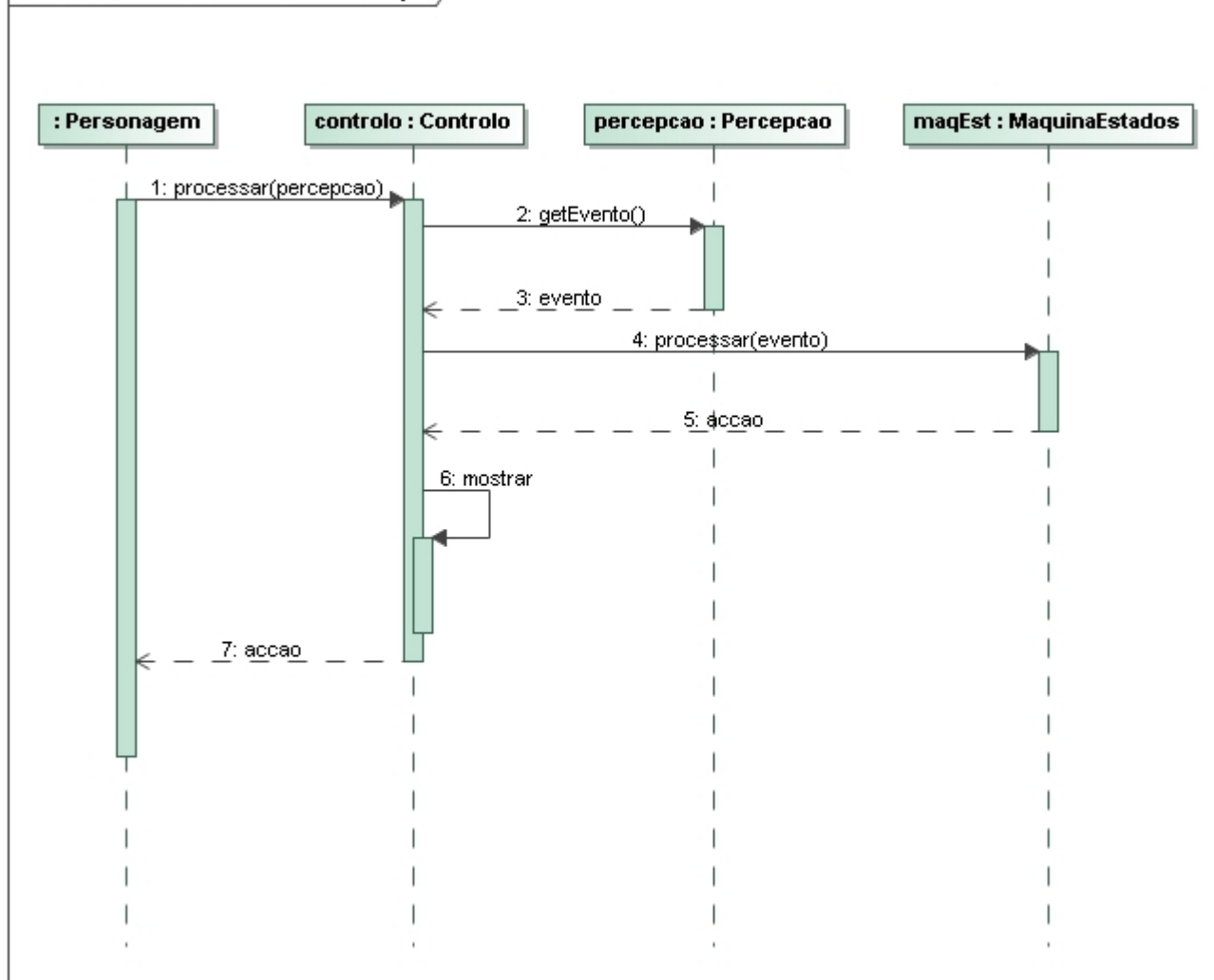
Controlo da personagem implementa a dinâmica do comportamento da personagem

IMPLEMENTAÇÃO...
(Estrutural)

PERSONAGEM

Processar percepção

Sequence Diagram [Processar percepção]



IMPLEMENTAÇÃO...
(Comportamental)

COMPORTAMENTO DA PERSONAGEM

Quando o jogo se inicia a personagem fica numa situação de **procura** de animais. Quando detecta algum **ruído** **aproxima-se** e fica em **inspecção** da zona, **procurando** a fonte do **ruído**. Quando volta a haver **silêncio** a personagem volta a uma situação de **procura** de animais. Quando detecta um **animal** a personagem **aproxima-se** e fica em **observação**. Caso o **animal** continue presente, a personagem **observa** o animal e fica preparada para o **registo**, se ocorrer a **fuga** do animal a personagem fica em **inspecção** da zona, à procura de uma fonte de ruído. Na situação de **registo**, se o **animal** continuar presente **fotografa-o**, caso ocorra a **fuga** do animal ou a personagem tenha conseguido uma **fotografia** do animal, a personagem fica novamente numa situação de **procura**.

Eventos do ambiente

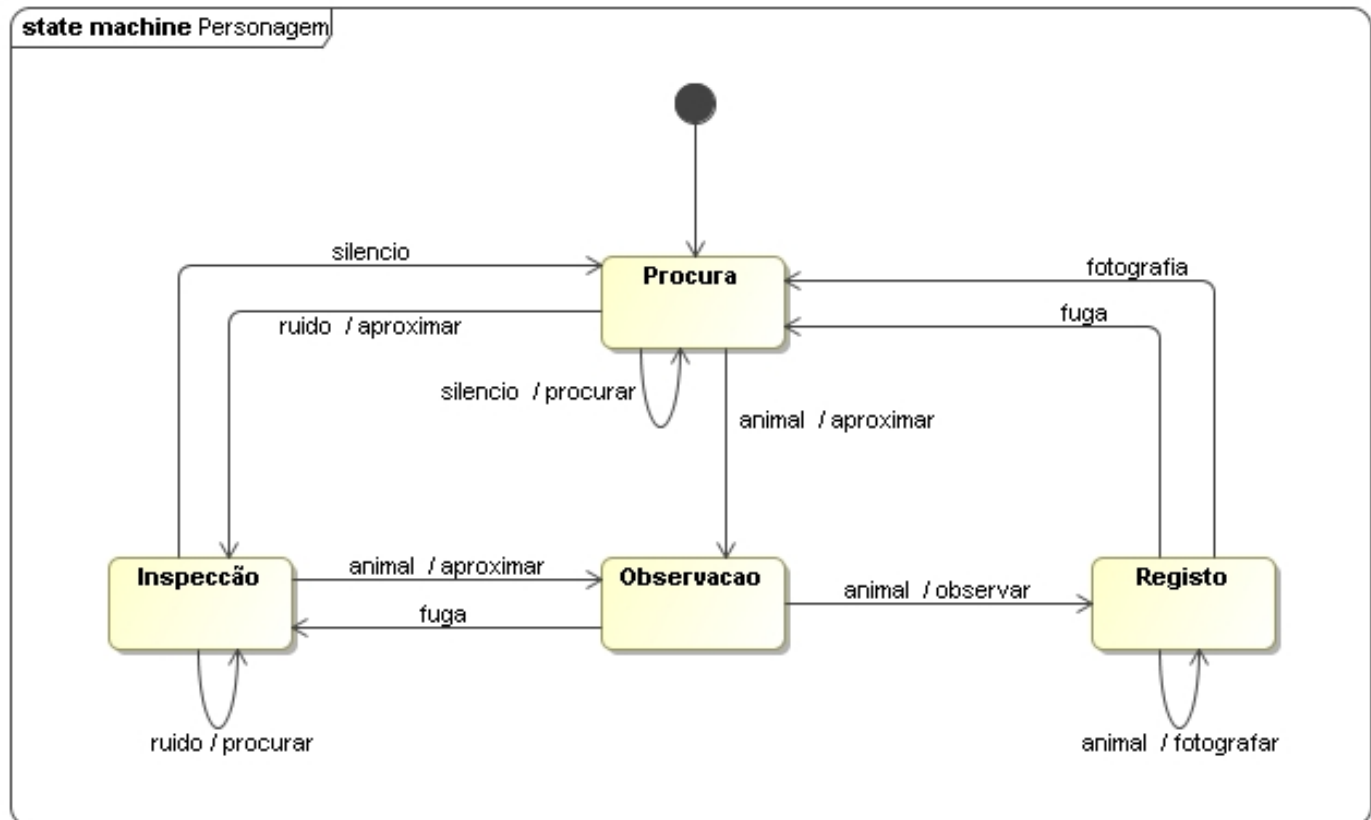
- Silencio
- Ruído
- Animal
- Fuga
- Fotografia
- Terminar

Acções da personagem

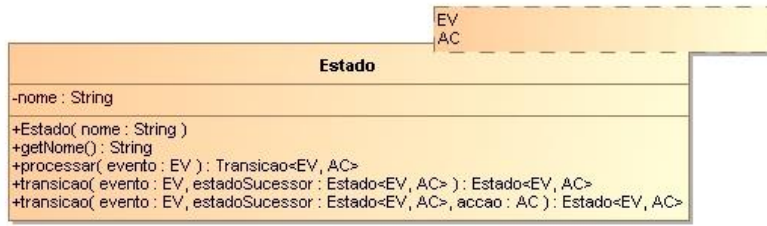
- Procurar
- Aproximar
- Observar
- Fotografar

Estados da personagem

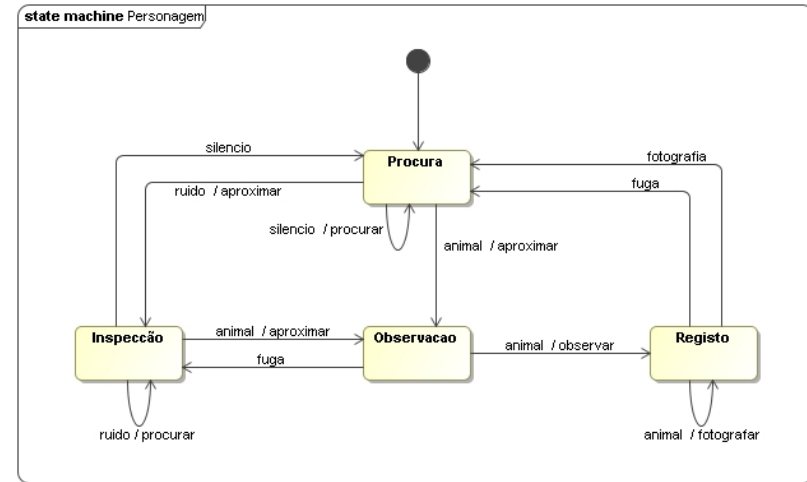
- Procura
- Inspeção
- Observação
- Registo



COMPORTAMENTO DA PERSONAGEM



Definição declarativa da
dinâmica do comportamento
da personagem



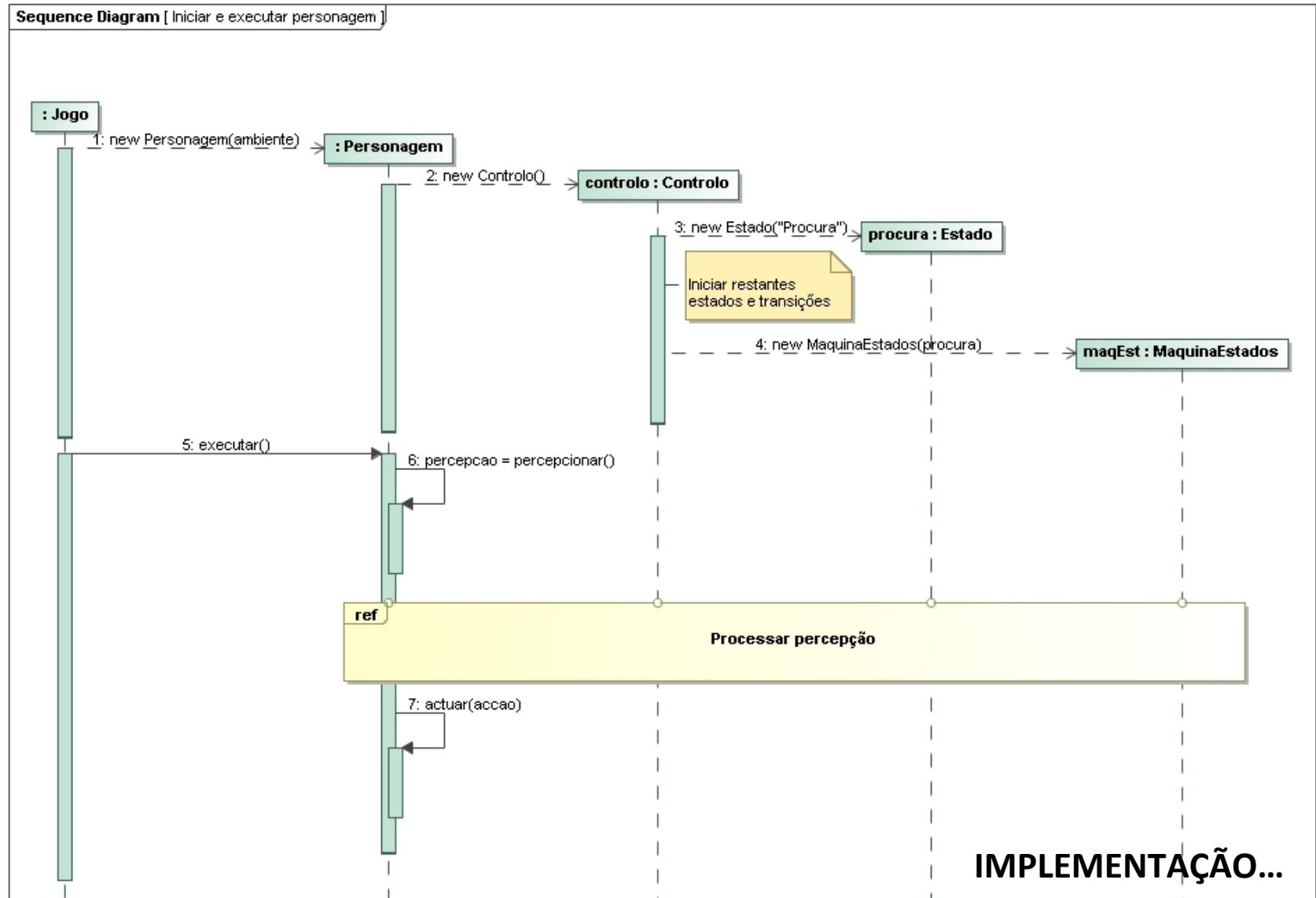
```
public Controllo() {
    // Definir estados
    Estado<Evento, Accao> procura = new Estado<>("Procura");
    Estado<Evento, Accao> inspeccao = new Estado<>("Inspeção");
    Estado<Evento, Accao> observacao = new Estado<>("Observação");
    Estado<Evento, Accao> registo = new Estado<>("Registo");

    // Definir transições
    procura
        .transicao(Evento.ANIMAL, observacao, Accao.APROXIMAR)
        .transicao(Evento.RUIDO, inspeccao, Accao.APROXIMAR)
        .transicao(Evento.SILENCIO, procura, Accao.PROCURAR);

    inspeccao
        .transicao(Evento.ANIMAL, observacao, Accao.APROXIMAR)
        .transicao(Evento.RUIDO, inspeccao, Accao.PROCURAR)
        .transicao(Evento.SILENCIO, procura);
}
```


COMPORTAMENTO DA PERSONAGEM

Iniciar e executar personagem



CONSOLIDAÇÃO TEÓRICO-PRÁTICA

- Consolidação do projecto
 - Verificar e corrigir erros
 - Rever e melhorar a clareza e organização do código
- Documentação
 - Documentar código realizado, *na sua relação com o suporte teórico e a arquitectura estudada*
 - Documentação requerida para avaliação
 - Classe *Estado*, todos atributos e métodos
 - Classe *Transição*, todos atributos e métodos
 - Classe *MaquinaEstados*, todos atributos e métodos
 - Classe *Controlo*, todos atributos e métodos